5 Exercises, week 5

The exercise solutions have to be returned at the latest on Sunday April 23'rd.

- Pen and paper exercises: you can scan the solutions and combile them into pdf or write them with Latex/word/... Compile all the answers into one pdf file
- Computer exercises: Report the answers to no-coding parts of exercises (if any) in pdf and compile with answers to pen and paper exercises. Additionally, send also the code used to solve the exercises. Note!
  - Only code should be returned. **Do not send data files**!
  - Write and comment the code so that it can be run by using your code only and the data provided in the course web pages.
  - If the lecturer cannot understand or run your code you will not get points from coding part even if the results were correct.
- zip all files into one folder to reduce the size of submission.

Send the zipped files to jarno.vanhatalo@helsinki.fi.

For basic properties and results concerning Gaussian distributions and processes see e.g.
`https://en.wikipedia.org/wiki/Multivariate_normal_distribution`
`http://www.gaussianprocess.org/gpml/chapters/`

5.1 Gaussian processes with binary outcomes

Load the data "binary_data.dat". This file contains binary data with coordinates $\mathbf{s}_i = [s_{i,1}, s_{i,2}]$ and outcomes $y_i \in \{0, 1\}$. We want to infer the data with a model

$$\mathbf{y} \,|\, \mathbf{f} \sim \prod_{i=1}^{n} \text{Bernoulli}(y_i|\pi(f(\mathbf{s}_i))) \tag{15}$$

$$f(\mathbf{s})|l, \sigma \sim \mathcal{GP}\left(0, k(\mathbf{s}, \mathbf{s}' \,|l = 0.5, \sigma^2 = 1)\right) \tag{16}$$

$$\tag{17}$$

where $\pi(f_i)$ is either the logistic, $\pi(f_i) = 1/(1 + e^{-f_i})$, or the probit, $\pi(f_i) = \Phi(f_i)$, link function. More specifically. We want to

- infer the posterior distribution of the latent variables, $\mathbf{f}$, at the training data locations
- calculate the posterior predictive mean and variance of $f(\mathbf{s})$ in a region around the data points and visualize it
- calculate the posterior predictive probability $\Pr(y(\mathbf{s}) = 1|\,\mathbf{y})$ in a region around the data points and visualize it

a) Infer the model using STAN.

b) Infer the model using the Laplace approximation described in Chapter 5 of Rasmussen and Williams (2006). For this you need to implement the algorithms 3.1 and 3.2 from the book.

c) Compare the posterior distributions for f_i obtained from the MCMC and the Laplace approximation at few training and test locations. Choose one location from near the $\Pr(y(\mathbf{s}) = 1|\,\mathbf{y}) = 0.5$ boundary and one from either $\Pr(y(\mathbf{s}) = 1|\,\mathbf{y}) \approx 0$ or $\Pr(y(\mathbf{s}) = 1|\,\mathbf{y}) \approx 1$ region.

Hint! The STAN warmup may fail to find good sampling parameters for the this model. To improve the tuning of the sampling parameters, you may want to try the following option for stan(...) function

```
control = list(adapt_delta = 0.99)
```

The Newton algorithm descriped in Algorithm 3.1 should work as such with these hyperparameter values and when initializing $\mathbf{f} = 0$. However, in some cases the Newton algorithm of Algorithm 3.1 (Rasmussen and Williams, 2006) does not converge even though the objective function was log concave. The reason is that the step size during the iterative algorithm is too long. If this happens, it helps to reduce the step size of the Newton algorithm. Hence, you should add the following into your algorithm after the line 8 of Algorithm 1 of Rasmussen and Williams:

```
i=0
  while (i < 10 && (logq_new < logq  || is.nan(sum(f)))) {
    # reduce step size by half if the Newton step is too long
    a = (a_old+a)/2;
    f = K*a;
    lpq_new = -t(a)*f/2 + y*(dnorm(f)/pnorm(y*f))
    i = i+1
  }
```

Here `a_old` is the vector $\mathbf{a}$ from the previous Newton round.